

# ZX-Appeal

\$1.50

Newsletter of the

## VANCOUVER SINCLAIR USERS GROUP

\*\*\*\*\*

\*\*\*\*\*

\* **NEXT MEETING:** \*

\* ----- \*

\* \*

\* KILLARNY COMMUNITY CENTRE \*

\* 6260 KILLARNY ST., VANC \*

\* \*

\* SEPT. 12, 7:00PM \*

\* \*

\* THIS WILL BE OUR FIRST MEETING \*

\* AFTER OUR SUMMER BREAK \*

\* \*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

### \*\*\*INSIDE\*\*\*

THIS ISSUE.....	2
BITS AND PIECES.....	2
PROGRAM READER.....	3
INTERRUPTS.....	4
MUSIC.....	5
TIPS.....	6
FAT LETTERS.....	7
PASCAL.....	8
COMPUTER FAIR.....	9
FANCY TITLES.....	10
MEMSCAN.....	11
HACKSEL.....	12
1000 SCROLLING.....	13
1000 SCROLL AGAIN.....	14
QUICK TAPE.....	15



**VANCOUVER  
SINCLAIR  
USERS  
GROUP**

**Happy  
Fourth  
Birthday**



-and they said it wouldn't last!

This Issue.....

We probably all thought that it wouldn't end but the sad fact is that SUMMER IS OVER! The darkness closes in a little earlier each evening and the nights are getting cooler. What this means is it's time to put away the barbeque and get down to some serious computing. Before I forget — HAPPY BIRTHDAY!! Twas September 8/82 when 8 of us met up in Al W.'s office to see if we could get a club together. Well, we're still here and going as strong as ever.

This issue is a potpourri - odd bits and pieces from here and there. I hope our exchange partners don't mind but I'm borrowing quite heavily from other newsletters now and then. Of course the other groups can use any material from our newsletters at any time. This way the many thousands of Sinclair and Timex nuts across the land and over the border can all share the very best of all the newsletters. I'm even able to include a couple of items from the newsletter of the Users Group in Mexico thanks to the translation services of a friend of Ken A.

Bits &  
Pieces.....

...Sorry about that:--It is quite possible that the video monitors I mentioned last issue are not compatible with the 2068. Marie K. bought one but found some screen distortion. A replacement unit also exhibited the same problem. Make sure you are able to return the unit if buying one. R & P are usually pretty good about this sort of thing.

--The listing for the graph plotter program in last issue didn't print clearly on one or two copies. The faded lines are:

```
165>LET O=0: IF K=2 THEN LET O=
1
170 DRAW OVER O;S,Z+(100/M)+Z8
172 IF Z<=0 THEN PLOT OVER O;X+
K+P,49
175 NEXT K
```

--The "2068 Tips" in last issue were for the SPECTRUM.

...if anyone is looking for Radio Shack TP10 paper and can't find any, here's why. HAV-INFO, the database service, is now marketing their dumb terminals. They bought all the TP10 printers and paper they could find. More is on the way and this time the RS dealers will be holding stock back for their own customers.

...if you are presently using an 80 column Epsom-compatible printer but wish you could afford one of the new Near Letter Quality printers then read on.

Dresselhaus Computer Products of Glendora, California, is marketing a neat little hardware add-on that not only allows NLQ on any Epsom or Epsom-compatible machine but also panel button font selection, switch selectable Epsom or I.B.M. character set, and all Graftrax-Plus features. Price? --\$80.00US. Better than buying a new printer!

...Karl B.'s robot course is again coming up at V.V.I. The first class will be on Tuesday, Sept.23 at 6:30 to 9:30. The course runs 10 weeks and will set you back \$135.00. Not bad when you consider you'll end up with a fully funtional, completely programable, mobile, talking robot as well as 30 hours of instruction on both theory and pratice of robot design and construction. If you'r interested in taking this course then hurry down to register NOW as the seating is very limited and it's first come, etc.

Karl tells me that BYTE magazine has accepted and paid for his article on "Building Your Own Robot" but has yet to tell him when it will appear print. Speaking of robots: I finally, with the help of Wilf R., came up

with a way to program my robot utilizing the 2050 modem serial port add-on from ZEBRA SYSTEMS and the Mterm I software. Now the house is full of the pitter-patter of little wheels.

I wanted to mention something else but it's slipped my mind. Next time then. Now I remember -----  
-----RENEW YOUR MEMBERSHIP!!



\*\*\*\*\*

## Program Reader

Have you at one time or another forgotten the name of a particular 1000 program but didn't want to wait the 10 minutes to see what it was? Help is at hand! This routine will read and print the name the program was saved under, all in a few seconds and without affecting the saved program.

Type in the program, remembering the 75 character REM statement. Run the program. After running the program, delete lines 10 to 70 and add line 10 Rand Usr 16514.

To use the routine just run the program and start the tape recorder. A few seconds after the program load pattern appears, the program will stop with the program name on the screen.

This would be a good routine to keep in the 8K NVM you built from last issues Hardware project instructions. When typing in the program just change line 20 to whatever address in the 8-16K area you wish to use. Then NEW the machine and call the routine whenever you like by simply RAND USR X, X being whatever address you chose for X in line 20.

1 REM SET UP REM STATEMENT WITH AT LEAST 75 SPACES.

10 LET A\$="CD230FCD8A4018FB0E0106003E7FDBFED3FF1FD2A2031717381110F1F1CD8A40CB7A792001D71730F4181DD51E94061A1DD8FE17CB7B7B38F510F5D12004FE5630C83FCB1130C3C9C95"

20 LET X=16514  
30 IF A\$="S" THEN STOP  
40 POKE X,16\*CODE A\$+CODE A\$(2)-476

50 LET X=X+1  
55 PRINT A\$( TO 2);" ";  
60 LET A\$=A\$(3 TO )  
70 GOTO 30

CD	23	0F	CD	8A	40	18	FB
0E	01	06	00	3E	7F	DE	FE
D3	FF	1F	D2	A2	03	17	17
38	11	10	F1	F1	CD	8A	40
CB	7A	79	20	01	D7	17	30
F4	18	1D	D5	1E	94	06	1A
1D	DB	FE	17	CB	7B	7B	30
F5	10	F5	D1	20	04	FE	56
30	C8	3F	CB	11	30	C3	C9
C9							

I have quit a few neat little machine code routines I keep in my HUNTER BOARD NVM. Each month I'll give you a couple you can try out and perhaps keep in your NVM.

This article is reprinted from the August/86 edition of "SMUG BYTES", the newsletter of the Sinclair Milwaukee Users Group.

## INTERRUPTS

by Lloyd Dreger

A discussion of interrupts was a bit too advanced a subject to cover in detail in my book "Introduction to 2068 Machine Code," since the beginning student has enough other things to learn, so here is a short discussion of them.

There are two kinds of interrupts available for the Z80 CPU, the maskable (MI) and nonmaskable (NMI) interrupts.

As the name implies, the maskable interrupt can be masked, which is another word for defeated. Machine code students will remember the disable interrupt (DI) and its opposite, the enable interrupt (EI). These commands only work on the maskable interrupt. Since a maskable interrupt comes along every 1/60th of a second, the Sinclair computers use this signal to refresh the screen. In addition, the 2068 also uses this signal to check the keyboard for an input. The ZX81/1000 computers have a SLOW (refresh the screen) mode and a FAST (don't bother with the screen) mode, which obviously are but little more than enabling or disabling the maskable interrupt. Certain machine code routines, such as data or code transfers, must be done without an interrupt until finished. Forgetting to do an EI before coming out of code back to BASIC results in a dead keyboard, which is equivalent to a crash.

HALT is another instruction which requires an interrupt to start the computer again. It works with either type of interrupt. However, since the MI can be disabled with the DI statement, the 1/60th second delay may not work on many programs. The NMI, on the other hand, always gets through. However, a word of caution is in order so read on.

What really happens with a MI is that all the registers are saved and the machine jumps to address 56 (38H) and follows the instructions given there. When a return is encountered, the registers are all restored as the machine goes back to working on whatever it was doing before the interrupt occurred.

The NMI is a hardware or peripheral interrupt. A line inside the computer called NMI normally has 5 volts on it. If for any reason it temporarily goes low, like being grounded, a NMI has occurred. The registers are again all saved along with the status of the MI as the machine jumps to address 102 (66H). On the ZX81/1000 machine it checks for SLOW and continues without any provision for adding another NMI routine. On the 2068 we have a problem as it checks for an address in the system variable located at address 23728-23729. If you check your 2068 Users Manual you will find that these addresses are not used. This is due to a bug in the next instruction in the ROM (whether this bug is deliberate or not is open to some debate). The instruction in effect reads "jump to the address only if it is 0000," effectively doing a wipeout or restart. If for some reason you poked something into this address, the NMI is ignored and a return to the program is made.

This is NOT the way an NMI should work. The errant instruction at address 109 should be JR Z, not JR NZ. This change would cause the computer to return immediately if the NMI interrupt address at 23728/9 was 0000 or jump to the address contained there and handle the interrupt with a routine you could write, finally returning to the program when finished. Unfortunately, the error is in ROM and nothing short of burning a new ROM on an EPROM will permanently correct it. Unless . . .

Unless you are NOT in the home bank. If you are in the dock bank with an LROS program, your technical manual states



For example, most printers have a 2k to 8k buffer which is loaded and then printed out. When empty, it is reloaded with the next batch, etc. All the time that the printer is emptying the buffer it is sending a signal saying "I'm busy, don't send more yet." The way it stands at present, the driver program monitors this signal in a wait loop and is really wasting time. Instead the "busy" signal could be put on the NMI line thus freeing the CPU to do something else and return for another buffer reload only when needed.

Or for another example. Your modem is monitoring the phone for calls. Nothing might happen for hours, but when it does the modem needs attention immediately. Or yet another, your computer is set up to play watchdog for your house, yet you would like to do an iterative program that takes hours to run overnight and have the answer in the morning. You could do both with a working NMI. Sinclair had the right idea with the NMI routine address at 23728. This routine with the same error is also in the Spectrum ROM. Why did they chicken out?

\*\*\*\*\*

" TURN OFF THE BUBBLE MACHINE "

Well it's not Lawrence Welk but it is 'music' on the 2068. This program uses just the BEEP command. If only someone would come up with a friendly easy-to-use, full-blown 3-part harmony music program utilizing the full potential of the 2068. That I would like to see!

\*\*\*\*\*

```

50 GO TO 50
10 DATA 19,18,18,15,19,-12,16,
12,15,11,16,-5,12,7,11,7,12,-12,
7,4,7,4,7,-5,4,0,5,2
15 DATA 7,-12,9,5,11,2,12,4,14,
-5,16,7,17,-10,14,9,14,7,14,-5
20 DATA 17,14,16,13,17,-10,14,
11,13,10,14,-5,11,7,10,6,11,-10,
7,2,7,-1,7,-5,19,11,18,10
25 DATA 19,-10,21,17,19,16,17,
-5,16,11,14,5,12,-12,12,-8,12,-5,
12,0
30 DATA 14,9,14,7,-6,14,9,6,9,
4,-10,9,12,4,12,6,-5,12,11,7,11,
-10,11,-5
35 DATA 14,11,14,7,14,-6,9,6,9,
4,9,-10,12,6,9,2,11,-5,9,2,7,6,
7,-1
40 DATA 14,9,14,7,14,-6,9,6,9,
4,9,-10,12,6,14,6,12,-5,11,7,11,
-10,11,13
45 DATA 14,9,14,7,14,-12,16,7,
14,7,12,-6,11,7,9,0,7,-5,7,-1,7,
2,7,5
50 LET co=96: LET pl=0
55 FOR n=1 TO co
60 READ a,b
65 FOR c=1 TO 3
70 BEEP .02,a: BEEP .02,b
75 NEXT c
80 NEXT n
85 LET pl=pl+1
90 RESTORE : LET co=48: GO TO

```

This article is reprinted from the June/86 edition of the "THE DATA EXPANSION", the newsletter of the T/S Users Group of Fort Worth, Texas

TIMEX TIPS  
By Chuck Dawson

QUESTION: I have a Byte-Back Modem and would like to download programs for Bulletin Boards which have been sent in HEX. Is there any way?

ANSWER: The software supplied with Byte-Back modems will not translate Hex, so a good way would be to download the program into the buffer and then convert it after signing off. This is not as easy as it sounds, but it is possible. Below is a program that can be loaded from tape, then the terminal program activated by using RANDOMIZE USR 61187 (which does not wipe out the buffer.)

When using this or any other buffer download, it is important to remember that the buffer must be opened just before receiving the program. That is, avoid extra characters captured in the buffer ahead of the first byte of the download. Some BBS's automatically open your buffer for you by sending a Control-R (this works for M-TERM, Z-TERM 64, or the Byte-Back S-TERM software.) Other BBSs allow time for you to open the buffer before beginning the download. A few extra bytes at the end before closing the buffer does not seem to make any difference. Here is the program:

```
1 LET VARS=PEEK 23627+256*PEEK 23628: LET J=27128: PRINT
#1;'STANDBY'
2 FOR I=J TO VARS-2 STEP 2: IF PEEK I<48 OR PEEK I>71 THEN LET
I=I-1: GOTO 6
3 LET A=PEEK I: LET B=PEEK (I+1)
4 LET A=A-48-7*(A>57): LET B=B-48-7*(B>57)
5 POKE J,16*A+B: LET B=J+1
6 NEXT I
7 LET A=INT (J/256): LET B=J-256*A
8 POKE 23728,B: POKE 23729,A: POKE 23627,PEEK 23728: POKE
23628,PEEK 23729: CLEAR
9 STOP
```

Now, to check for accuracy, PRINT PEEK 23627. If you have not made any errors, you should see 5. Before SAVEing to tape, POKE 23627,6: POKE 23641,7. You can now SAVE the program to tape to be LOADED before each download. After the download is

completed, exit to BASIC. GOTO 1 to translate the program to usable form. The word STANDBY should appear at the bottom of the screen while the translating is taking place. When you see the program has been completed, look at the listing. If the downloaded program has the lines in the range of 1-9, there will be duplicates since the translator has lines 1-9. Do not use the DELETE key because this will delete all lines numbered 1-9. Instead, type 1 and ENTER. Continue until the first 9 lines are gone and just the downloaded program remains. Before trying it out, SAVE it to tape. By the way, don't expect to get a good download every time, but keep at it and you will be successful.

QUESTION: I am new to machine code programs and I would like to ask a basic question: If I call a short MC routine from within a BASIC program, how do I get the result back to a BASIC variable so I can use it?

ANSWER: This question keeps being asked, so I am glad to try to help. There are two ways to approach this problem. When a number has been manipulated by the various registers, it can be stored for later use by loading it into an unused memory location (or more usually two). After returning to BASIC, the number can be recovered by `LET A=PEEK loc + 256*PEEK (loc+1)`. Several numbers could be calculated by the MC part of the program and stored in different locations. All you have to do is remember where you put them. If the MC routine produces only one answer to be transferred to the BASIC program, a short-cut method is to leave the answer in the BC register (TS-2068 and Spectrum) or the HL register (TS-1000 or TS-1500). If the routine was called by the line `LET a=USR (64000)`, then when the MC routine RETURNS control to BASIC, the value that was in BC will now be in the BASIC variable A. If you called the routine with `PRINT USR (64000)`, then the answer will be printed on the screen upon RETURN

QUESTION: I have a problem with a program which includes user defined characters. It works just fine in the normal mode, but when run with the OS-64 (64 column cartridge), the user characters do not work.

ANSWER: When the TS-2068 is switched into one of the enhanced display modes like the 64 column, the user defined graphics are moved from 65386 to 63256. If you have defined a character for GRAPHIC A, for example, by POKING values into addresses starting at 65386, then you have poked the wrong location when in enhanced mode. It should have been 63256. You would have to change that location each time you switch from one mode to the other. A good way out of this is to use the built-in GRAPHIC ADDRESS CALCULATOR. That is, calculate the starting address of GRAPHIC A by the expression `LET address=USR 'A'`. Then you can POKE address and address+1 and address+2, etc. No more changing the program by hand each time you plug in or out the 64 column cartridge. By the way, if you like to put machine code in a REM statement at the beginning of your program, it, too, will move when in enhanced mode. Always use `PEEK 23635+256*PEEK 23636` to calculate the beginning of the program listing. Add five to get the correct USR call location if the MC starts just after the REM keyword in the first line of the program.

\*\*\*\*\*

This program was developed by Dick Scoville of the Triangle Sinclair Users Group for the 2068. The program goes through the display file and darkens up each pixel. The resulting program listing printout is nice and dark - especially when photocopying.

```
10 CLEAR 65506
15 LET a=65507
20 READ n: POKE a,n
30 LET a=a+1: GO TO 20
40 DATA 17,0,221,213,1,0,3,42,
54,92,36,126,167,31,182,18,35,19
,13,32,246,16,244,225,37,34,54,9
2,201
```

The following is reprinted from June/86 edition of "Expansion", the newsletter of the T/S Users Group of Fort Worth, Texas.

## PASCAL

By David Baulch

One of the many reasons that I wanted to own a personal computer was for wordprocessing, grades (Mr. Baulch is a teacher - Ed.), keeping records (database), and for learning. I learned from a book with questions to the proper people to help me over the tight spots. I was told many times, "The only way to learn to write programs is to write programs." I never knew there was to be a plan or flow-chart. After learning BASIC, I found there were many other computer languages. One of these was PASCAL. PASCAL, I discovered, had the structure I was missing in my earlier learning. I had to go back and re-learn all over again.

Dr. Niklaus Wirth, inventor of PASCAL in 1968, wanted a teaching aid to be used as a hypothetical language to teach others programming and structured, top-down design. Thus, PASCAL was born of a necessity. By 1970, he had a working implementation. After using PASCAL in the commercial and academic world for a few years, he found many shortcomings, many nonstandard extensions and awkwardness for large programs, and defects, problems in manipulating the memory and peripherals in a computer directly. He went further to form MODULA and, finally, MODULA-2. Even though Dr. Wirth found PASCAL to be lacking, many people continue to use it and praise its virtues.

"Introduction to PASCAL and Structured Design", D.C. Heath and Company, Nell Dale and David Orshalick, is a very good book either for the beginner or for one who is more advanced in the use of computers and programming languages. This is a book I borrowed from the Computer Science classes at the school where I teach. Most of the time when you hear the term PASCAL, you think of Borland's Turbo PASCAL. If you are just beginning, PASCAL sounds like some type of musician or an artist of some sort. Of course, neither is true.

Computer programming is nothing more than the process of planning a sequence of instructions for a computer to follow, and the program is the sequence of instructions outlining the steps to be performed. There are two phases to the programming process: the problem-solving phase (analyzing the problem, developing the algorithm, and testing), and the implementation phase (testing the program, debugging the program, and using the program). Utilizing the top-down design approach, you break the problem into a set of sub-problems until it has expanded every task to the smallest detail (hierarchical or tree structure). This makes the problem easier to handle. Once the problem is



broken down to smaller problems, the task of encoding your program becomes a relatively simple procedure.

One of the nice things about PASCAL is the statements that are used in the construction of a program. The heading gives you the name of the module or program and what is expected to be used. The declarations are like LET statements and can be broken up into things like constants, variables, characters, and Boolean. The statements are the executable part of the program or module, and are set off with a "BEGIN-END" pair. This makes the documenting of the program something that even a non-programmer can read. Many times the declarations and statements are self-documenting. Once the program is written and debugged, it is compiled - because PASCAL is a compiled language. Therefore, when the code is RUN, it is run as assembly code.

I still have much to learn, but it seems to be somewhat easier learning PASCAL after learning BASIC.

The following is extracted from a recent West Coast Computer Society Newsletter:

PACIFIC COAST COMPUTER FAIR  
ASSOCIATION (PCCFA)

At its April 14th meeting the PCCFA voted to take some drastic steps that will allow it to continue to put on events in this and future years.

Instead of a flashy, expensive, 2-day event at Robson Square such as in past years, future fairs are envisioned to be simpler 1-day shows, with a swap-meet/flea market (like the one at S.F.U. in January) as an integral feature.

Why the change? Firstly the fair was organized mainly by volunteers in their 'spare time'. The people who have put in most of the work (often during business hours) to put on past fairs, can no longer do so, and not enough new faces have stepped forward to replace them. the new format will only take a fraction of the effort.

Secondly, the losses experienced in the past two years have reduced the financial reserves to the point where putting on another large fair would be difficult and risky. But we should be able to put on the simpler fair once or maybe even twice a year for years to come.

Thirdly, home computers are no longer the novelty they were 5 or 6 years ago. As a result, fewer people are looking to buy their first computer, and more are looking for software, peripherals, and accessories (especially bargains) to use with the computer(s) they already have. The addition of the swap-meet/flea-market section caters to these people. Finally, the reduced costs of the new fair will allow a reduction in the admission charge.

The date of the next fair has not been finalized, but will likely be sometime in September.

# FANCY TITLES From YOUR SINCLAIR and ZX COMPUTING

Have you ever wanted to have the title of your own personal program to have something special? The people from 'across the pond' have come up with some interesting variations for you to try. Since these are so short, I suggest that you just type them in and give them a try. Then, change 'things' around and experiment with them. I have been 'playing' and they are quite interesting.

From Eddie Duncan-Dunlop, Bridgend, Mid Glamorgan, a program that 'shoots' titles onto the screen with a sort of Laser Effect.

```
1 REM YOUR SINCLAIR
10 REM Shooting Titles
20 REM By Eddie Duncan-Dunlop
60 PRINT AT 0,0; INK 7; ' YOUR SINCLAIR'
70 FOR x=0 TO 127
80 FOR y=0 TO 7
90 IF NOT POINT (x,y+168) THEN GOTO 150
100 PLOT 2*x,3*y+80
110 PLOT 0,0: DRAW OVER 1; 2*x,3*y+79
120 PLOT 0,0: DRAW OVER 1; 2*x,3*y+79
130 BEEP .001,30
140 PLOT 2*x,3*y+81
150 NEXT y
160 NEXT x
```

TWO VARIATIONS: the first will invert the title and the second will 'darken' the effect. Just substitute either or both lines to see what they will do.

```
90 IF POINT (x,y+168) THEN GOTO 150
AND/OR
70 FOR x=0 TO 127 STEP 1/3
```

Jasper Visser sent in this short routine to print in a rather original style on the screen. From ZX COMPUTING:

```
1 REM characters
100 LET I=0: LET h=0
130 INPUT 'A word please...'; LINE a$
140 IF LEN a$>10 THEN PRINT AT 10,0; FLASH 1; BRIGHT 1; 'I - 10
CHARACTERS !!': PAUSE 50: PRINT AT 10,0;,,, GO TO 130

150 PRINT AT 0,0; INK 7; a$
160 FOR a=0 TO LEN a$-1
170 FOR b=175 TO 168 STEP -1
180 IF POINT (a,b)=1 THEN GO SUB 220
190 LET h=h+3: NEXT b: LET h=0: LET I=I+3: NEXT a
200 PRINT AT 0,0;,,,
210 STOP
220 PLOT 1,50-(h+3): IF POINT (a-1,b)=0 THEN DRAW 0,3
230 PLOT 1,50-h: IF POINT (a,b+1)=0 THEN DRAW 3,0
240 PLOT 1+3,50-h: IF POINT (a+1,b)=0 THEN DRAW 0,-3
250 PLOT 1+3,50-(h+3): IF POINT (a,b-1)=0 THEN DRAW -3,0
260 RETURN
```

\*\*\*\*\*

EBZUG      PUG      SVSTUG

by Russell English

It works by scanning every other address, checking for zeros. If something other than a zero is found, it assumes that the address is being used and a point is plotted on the screen.

```

To scan every byte in the
first half of memory, POKE
65519,0. To scan every byte in
the second half of memory, POKE
65502,128: POKE 65519, 0. For
SPECTRUM, POKE 65515,229 : POKE
65516,34.

```

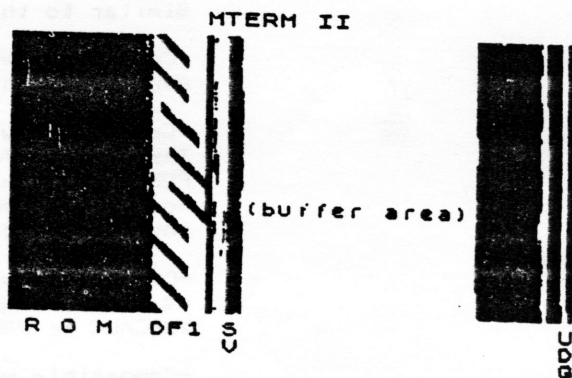
A title can be added to a display with: `PRINT AT 1,10;"name"`. The display may be compared to the memory map on p. 254 of the 2068 User Manual.

```

10 FOR x=65500 TO 65533
20 READ y: POKE x,y
30 NEXT x
40 DATA 33,0,0,14,1
50 DATA 6,27,62,0,197
60 DATA 229,190,40,3,205
70 DATA 52,38,225,35,35
80 DATA 193,4,62,156,184
90 DATA 32,236,12,62,255
100 DATA 185,32,228,201
110 RANDOMIZE USR 65500
115 STOP
120 SAVE "Memscan"CODE 65500,34

```

SPECTRUM POKES: #1=229; #2=34



## A NEW 2068 PRINTER INTERFACE!

-Works with all printers which use the Centronics parrallel standard.

-Includes all connecting cables. No other purcha es are required.

-Uses the LPRINT and LLIST commands directly from basic. No USR commands necessary.

-Allows for full page, high resolution graphic screen copys.

-Allows for full page, high resolution color screen copys using a grey scale system. Similar to the Macintosh screen.

-Compatible with ALL software designed for the Aerco Centronics Interface. e.g. MSCRIPT

-Compatible with ALL software designed for the Tasman Interface. Including Tasword II. (Some POKes are necessary to customize Tasman Software to the Hacksel Interface. These are included of course.)

-Compatible with a great many programs that use the LPRINT and LLIST commands such as VU-CALC or PRO-FILE.

-Compatible with Omnicalc II.

-Quite simply the most compatible printer interface availiabile for the Timex 2068! And as we are a Canadian company, there is no exchange or duty. Ordering a Tasman Interface from the States would cost well over \$100.00!

-Availiabile in two types of boards.

CARTRIDGE BOARD	\$79.99
REAR EDGE CONNECTOR BOARD	\$75.99

Please add \$4.00 Postage and Handling.  
Ontario Residents please add 7% sales tax.

Send Cheque or Money order to:

# Hacksel Electronics

247 QUEEN ST. WEST TORONTO ONT. CANADA M5V 1Z4 (416) 596-1663



## SCROLLING UTILITY FOR THE T/51000/1500

by David Novotnik

Taken from INFO-G.U.T.S. (Users Group of Mexico), Vol. 1, No. 4. Text explanation by Ken Abramson.

The following machine language program, once loaded, will be stored in a reserved section of RAM above RAMPOT, and will remain there, even if the NEW command is used. It can then be called into action by simple USR commands written into any BASIC program. Not only will the program scroll UP, but it will also scroll DOWN, LEFT, and RIGHT.

USER COMMAND	RESULT
RAND USER 32521	SCROLL UP
RAND USER 32542	SCROLL DOWN
RAND USER 32566	SCROLL RIGHT
RAND USER 32586	SCROLL LEFT

[illegible]

"SCROLLING"  
T/S 1000/1500  
BY BRIAN LITTLE

Load the program into the computer in the normal way. Then RUN it. It should end up with a "K" cursor on the screen. The SCROLL UTILITY is now installed in RAM, but it cannot be listed.

Now type in the following little test program and run it. You should see some scrolling action take place when the FOR-NEXT loops are processed (these loops control the number of steps the screen will scroll in each direction).

```
1 REM SCROLLING TEST PROGRAM
  (USE ONLY AFTER LOADING
  THE SCROLLING UTILITY
  PROGRAM ABOVE)
```

```

10 FOR N=1 TO 255
20 PRINT CHR$ N;
30 NEXT N
40 REM SCROLL UP
50 FOR U=1 TO 10
60 RAND USR 32521
70 NEXT U
80 REM SCROLL DOWN
90 FOR D=1 TO 10
100 RAND USR 32542
110 NEXT D
120 REM SCROLL RIGHT
130 FOR R=1 TO 10
140 RAND USR 32586
150 NEXT R
160 REM SCROLL LEFT
170 FOR L=1 TO 10
180 RAND USR 32586
190 NEXT L

```

```
1 REM      "HEADSTAND"
           T/51000/1500
           BASED ON A PROGRAM
           BY J. REYNA
           MEXICO
```

[illegible]

Here's an oldie that first graced our pages back in the spring of '83, originating in Sinclair Programs Jan/Feb '83.

**S**ROLL is a machine code program to move the display across the screen in any one of eight directions. Enter Listing 1, keying 126 characters in the REM statement in line 1. The variables S1 to S9 correspond to the compass points NW, N, NE, W, E, SW, S and SE. RUN the program and the machine code will be POKEd into the REM statement. Then delete lines 10 to 70 and replace them with the demonstration routine in Listing 2. If the direct command "GOTO 10" is then entered, the versatility of this scroll program will be demonstrated. Submitted by Stephane Crainic, of Paris.

[16K ZX-81]

## LISTING 1

```

1 REM 1111111111112222222222223333
3333333444444444555555555555666666
555555667777777777888888888888999999
999900000000000011111111111122222222
23333333
2 LET S1=16526
3 LET S2=16514
4 LET S3=16633
5 LET S4=16560
6 LET S6=16588
7 LET S7=16619
8 LET S8=16629
9 LET S9=16612
10 LET A$="2A0C40E511210019D10
10502EDB0C92A1040114300ED052E5112
100ED52D101B502EDB82A0C400620233
60010FB0C92A0C4011D0621906162B4E3
6002B7EFE762602180310F2C9714F18F
12A0C400616234E35080237EFE7628021
80310F2C9714F18F1CDCC40CD9140C9C
D0040CD9140C9CDB040CD8240C9CDCC4
0CD8240C9"
19 FAST

20 LET I=16514
30 FOR J=1 TO LEN A$ STEP 2
40 POKE I, (CODE A$(J)-26)*16+C
ODE A$(J+1)-28
50 LET I=I+1
55 IF PEEK I=118 THEN GOTO 70
60 NEXT J
70 SLOW

```

LISTING 2

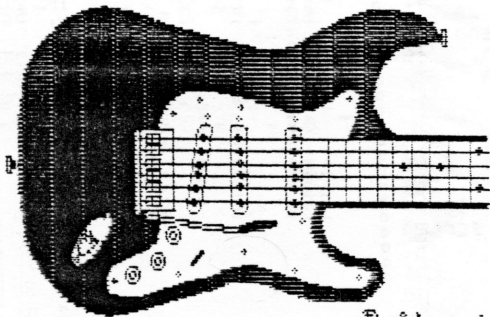
```

10 LET A$="SQ"
20 LET A$(2)=STR$(VAL A$(2)+1)
30 IF A$(2)="5" THEN GOTO 20
40 FOR I=1 TO 22
50 PRINT "====SINCLAIR USER AND
PROGRAMS===="
60 NEXT I
70 FOR I=1 TO 32
80 RAND USR VAL A$
90 NEXT I
100 CLS
110 GOTO 20

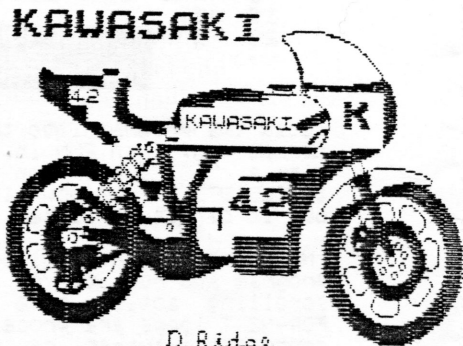
```

\*\*\*\*\*

Here are a few examples of the type of graphics that can be created with ARTWORX. Most of this Newsletter was put together with ARTWORX. If you own a 2863 you have to get ARTWORX!! It's that simple.



D. Shaggi



D. Ridge.

Another goodie from the mists of time. This also appeared in our pages in the Spring of '83 and comes from Sinclair Programs.

**Q**UICK TAPE, a machine code tape routine, will allow you to save and load 16K programs on the ZX-81 in less than a minute. In technical terms the program increases the machine baud rate to 1,500. To put in the program for the first time, type-in and run the hex loader, which is the first program in the listing. In answer to the program prompt, type-in the machine code listing line by line. When you have finished the listing, type 'S' and the machine code will be transferred to the REM statement in line 1. Then delete the loader, line by line, keeping the REM intact.

Type-in the second program, starting at line 10. The REM in line 1 should look like the REM statement which you have just created. When you run this program it will SAVE itself on to tape and then NEW automatically. The machine code routine is then stored above RAMTOP ready for use.

To SAVE a program, type RAND USE 32512 and to LOAD, type RAND USR 32525. When you want to use the program LOAD it normally and RUN. The Basic will destroy itself after transferring the machine code above RAMTOP. The routine will then be ready to use on your faster computer.

Quick Tape was sent by K S Beddoe of Botley, Southampton and is proving very useful.

```

1 REM 244 Q'S
10 LET A$=""
20 INPUT B$
30 IF B$="S" THEN STOP
40 LET A$=A$+B$
50 CLS
60 PRINT A$
70 PRINT "LENGTH= ";LEN A$/2;"
  BYTES"
80 GOTO 20
100 LET X=16514
105 FAST
110 IF A$="" THEN GOTO 160
120 POKE X,16*CODE A$+CODE A$(2
)-476
130 LET A$=A$(3 TO )
140 LET X=X+1
150 GOTO 110
160 SLOW
170 STOP

10 SAVE "FAST TAPE ROUTINE"
20 PRINT "FAST TAPE ROUTINE"
30 PRINT "TO SAVE RAND USR 325
12"
40 PRINT "TO LOAD RAND USR 325
25"
50 PAUSE 150
60 RAND USR 16738

```

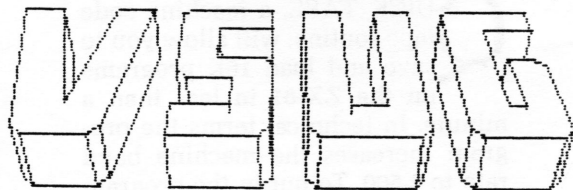
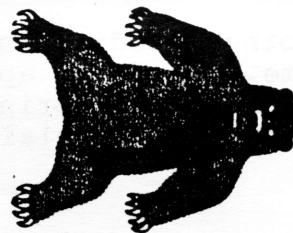
#### Machine code listing

16514	CD	23	0F	11	06	7F	CD	2B
16522	7F	CD	2B	0F	C9	CD	23	0F
16530	21	1D	7F	22	16	40	CD	70
16538	7F	CD	2B	0F	C9	0B	0B	0B
16546	00	00	00	00	00	00	CD	A8
16554	03	38	F9	EB	11	CB	12	CD
16562	46	0F	30	2E	10	FE	1B	7A
16570	B3	20	F4	CD	4E	7F	CB	7E
16578	23	28	F8	21	09	40	CD	4E
16586	7F	CD	FC	01	18	F8	5E	37
16594	CB	13	C8	9F	E6	02	C6	01
16602	4F	D3	FF	06	23	10	FE	CD
16610	46	0F	30	72	06	1E	10	FE
16618	0D	20	EE	C3	D8	7F	1B	E0
16626	CD	A8	03	CB	12	CB	0A	CD
16634	7C	7F	18	FB	0E	01	06	00
16642	3E	7F	DB	FE	D3	FF	1F	30
16650	49	17	17	38	28	10	F1	F1
16658	BA	D2	E5	03	62	6B	CD	7C
16666	7F	CB	7A	79	20	03	BE	20
16674	D6	23	17	30	F1	FD	34	15
16682	21	09	40	50	CD	7C	7F	71
16690	CD	FC	01	18	F6	D5	1E	31
16698	06	0E	1D	DB	FE	17	CB	7B
16706	7B	38	F5	10	F5	D1	20	04
16714	FE	56	30	B2	3F	CB	11	30
16722	AD	C9	7A	A7	28	BB	CF	0C
16730	A7	06	50	10	FE	C3	6E	7F
16738	21	82	40	11	00	7F	01	E0
16746	00	ED	B0	21	FF	7E	22	04
16754	40	C3	C3	03				



HELP: Does anyone have a copy of  
ZX Inlay? Mine won't  
load any longer and I  
would appreciate it if  
I could make a copy from  
someone.....Rod

GRIZZLY



VANCOUVER SINCLAIR USERS GROUP

THE VANCOUVER SINCLAIR USERS GROUP HAS  
BEEN IN EXISTENCE SINCE 1982. WE ARE A  
SUPPORT GROUP FOR THE OWNERS AND USERS  
OF THE: MICROACE, ZX80, ZX81, T/S/1000,  
T/S1500, SPECTRUM, SPECTRUM +, T/S2068,  
AND QL COMPUTERS.

PRES.--KEN ABRAMSON

V/PRES.-- ?

TREAS. & EDITOR--ROD HUMPHREYS

OUR MEMBERSHIP DUES ARE ONLY \$15.00/YEAR  
AND MAY BE SENT TO THE TREASURER.

ROD HUMPHREYS  
2006 HIGHVIEW PLACE  
PORT MOODY, B.C., V3H 1N5

MEMBERSHIP INCLUDES A SUBSCRIPTION TO  
ZXAPPEAL OUR MONTHLY NEWSLETTER.  
ZXAPPEAL ACCEPTS ADVERTISING, OUR  
\*\*PREPAID\*\* RATES ARE:

\$40.00 -- FULL PAGE

\$20.00 -- 1/2 PAGE

\$10.00 -- 1/4 PAGE

ZXAPPEAL HAS A PRINT RUN OF 100 COPIES  
PER MONTH FOR MEMBERS AND IS ALSO  
DISTRIBUTED TO APPROX 50 OTHER SINCLAIR  
USER GROUPS THROUGHOUT NORTH AMERICA AS  
WELL AS OVERSEAS VIA THE NETWORK.

NETWORK CORRESPONDENCE MAY BE DIRECTED  
TO THE EDITOR AT THE ABOVE ADDRESS.